Data Systems Technology Laboratory Series *DSTL-94-002*

Usability Testing Handbook

January 1994

National Aeronautics and
Space Administration

Goddard Space Flight Center

Usability Testing Handbook

January 1994

Prepared for the
Data Systems Technology Division
Software and Automation Systems Branch / Code 522
GODDARD SPACE FLIGHT CENTER

By
Dana L. Uehling

Copies of this document may be obtained by writing to:

Dana L. Uehling
Code 522.1
NASA/Goddard Space Flight Center
Greenbelt, Maryland 20771

Usability Testing Handbook

Approvals

Sylvia Sheppard Date
Head, Systems and Software Environments Section/Code
522.1

Martha Szczur Date
Head, Software and Automation Systems Branch
Code 522

Abstract

This document discusses the purpose of usability testing, what it is and what information it provides, at what point(s) in a project's life cycle to perform usability testing, how to prepare for and perform the usability test, and what to do with the results of the usability testing. It is not meant as an exhaustive guide for all ways to perform usability testing, but as a guide to the process we currently use in the Software and Automation Systems Branch (Code 522) at the NASA / Goddard Space Flight Center.

Keywords: Usability testing
human factors

Table of Contents

List of Figures and Tables

Figures

Acronyms and Abbreviations

GSFC	Goddard Space Flight Center
HCI	Human-Computer Interaction
NASA	National Aeronautics and Space Administration

1 Introduction: Why Do You Do Usability Testing?

A good user interface allows end-users to perform their jobs faster and easier. If the application's user interface gets in the way of the users performing their tasks (i.e., makes them lose their concentration), then the user interface is hindering end-user productivity. In fact, usability is the one element of an application that is most constantly apparent to the end-user.

When designing an application's user interface, how do you know if it is going to make the end-user's job easier or harder? There are numerous human-computer interaction (HCI) activities and usability evaluation methods that can contribute to development of a good user interface [5,6]. For instance, techniques that are found to be effective are: end user requirements definition, studies of end user work context, style guidelines, developer checklists, design walkthroughs, heuristic evaluations, prototype demonstrations, user beta testing, user surveys, and observational evaluations.

One of the difficulties with many of the usability evaluations is that they are subjective in nature (i.e., people see things differently) and it is difficult to measure the benefit of qualitative evaluations. A proven effective approach to identifying usability problems quantitatively is with laboratory-controlled usability testing. This involves observing test users (from the end-user community) perform prescribed tasks and gathering quantitative usability data about each task. The benefits and merits of performing structured usability tests on an application's user interface have been expounded upon in numerous papers written by HCI experts [3,4]. These benefits include: increased productivity, decreased training costs, decreased service costs, decreased user errors, decreased maintenance costs, and increased user satisfaction.

We in Code 522 have been involved with the design and development of user interfaces for many years. While we practiced the model of prototyping and getting users involved in reviewing the under-development user interfaces, we lacked a more formal way of evaluating and measuring whether the resultant user interfaces met the end-user's usability requirements. This led to usability measurement and testing.

While we in Code 522 have not yet conducted any cost benefit analyses associated with our usability testing, the metrics and user comments collected from our follow-up usability testing indicate that changes made to the user-interface as a result of the information gained by performing the usability tests allow end-users to perform their jobs faster and be more satisfied while performing their jobs.

This document describes a usability testing method that we have used and that has proven to be easy, cost effective, and useful. As we continue to refine and improve our usability engineering techniques, this handbook will be updated to reflect these enhancements.

2 When Do You Do Usability Testing?

Usability testing is most effective when introduced into the software development cycle before and after the implementation phase. With project leaders becoming more aware that fixing a problem early in the development cycle saves money [1,2], prototyping of the user interface is making its way into the requirements definition and design phases. Frequently these prototypes are demonstrated to, or used by, the end-users to get their reactions. While this approach is certainly a useful vehicle for getting valuable feedback, it can be misleading for identification of usability problems. Frequently, the end-user's remarks are not focused on performing a required task, but instead are personal opinions, which may change when performing specific functions in the operational environment. Usability testing of the prototypes is an effective technique for getting end-user feedback in a more structured atmosphere, with the observations and metrics providing an objective critique of the user interface.

On the other end of the development cycle, usability testing can play a role in the system testing phase. Hopefully, all the serious usability problems will have been identified during the requirements and design phases, but to validate that the operational system has a "good" user interface, structured usability testing on the most critical application tasks will ensure that the end-users are going to be helped in doing their jobs, rather than hindered by an awkward, poorly designed user interface.

3 How Do You Do Usability Testing?

Usability testing is a multiple step process which includes pre-test preparations, the actual testing, analysis of the test results, modifying the software system, and re-testing of the system. Each of these steps is discussed in the following section.

3.1 Pretest Preparations

The pretest preparations include defining the tasks you wish to evaluate, selecting performance goals for those tasks, and selecting the people who will participate in the usability test.

3.1.1 Task Definition

The first step in usability testing is to define what tasks you wish to test for your system. Except for the simplest of systems, it is not reasonable to test all end-user tasks, so a sub-set of tasks needs to be defined. When choosing tasks to test, choose those tasks with the following characteristics:

- most frequently used,
- critical or high risk, and
- complex.

In addition, include some simple tasks to provide the test users with a sense of accomplishment and to relieve stress on the test users.

When defining the tasks, break the tasks into logical steps. Be careful to define the tasks in terms of <u>what</u> to do (create a new item with the name "New item"), not <u>how</u> to do it (select "New item" from the menu and set the name field to "New item"). Your tasks should not refer to menus, fields, buttons, etc. You should use terms which are not system-specific.

Once defined, the tasks should be reviewed for clear definition and appropriateness. This review process should be done by someone who is familiar with the proposed use of the system, but is not a developer of the system. Developers of the system may overlook missing information or unclear wording of the tasks due to their in-depth knowledge of the system.

3.1.2 Metric Selection and Goal Setting

The next step is to define what it is you wish to measure while the tasks are being performed. The most common measures are:

- the time to perform the task and
- the "user satisfaction" (how easy or hard the user felt the task was to perform).

Another common measurement that can be used is the number of errors. The challenge with using this measurement unit is in defining what constitutes an error. While using the number of errors as a metric is not addressed in this guide, the same principles of measurement would be applied.

Once the metrics are selected, performance levels should be defined for each task. These performance levels include:

- the best-expected level,
- the worst-expected level, and
- the goal level.

First determine what the *best-expected* score should be for each task. For time to perform the task, this is determined by having one or more developers of the system perform each task and recording their time-to-perform. They will typically have the best performance on the system since they know it quite well.

The *worst-expected time* is typically considered to be six times the *best-expected time*. This calculated time should be reviewed, though, to see if it is reasonable. Is it really acceptable to have the performance at that level? Take into account the complexity of the task and time needed to speak aloud during the testing process.

Once the best- and worst-expected-times are determined, decide on what the goal time should be. Ask yourself, what performance level would you like to see achieved? At what level would you consider the user interface to be "usable"? This level should fall somewhere between the best- and worst-expected levels.

For user satisfaction ratings, a scale of 1-5 can be used, with 1 indicating that the task was hard to perform, and 5 indicating that the task was easy to perform. Typically you would select 5 to be your best-expect level, and 3 to be your worst-expected level. (You choose 3 instead of 1 as your worst since it is not acceptable if the user finds the task hard to perform.) You could choose a 4 or a 5 as your goal level, taking into account the complexity of the task.

3.1.3 Test Personnel Selection

There are several people who participate in the usability testing process. These people are:

- the test users,
- the facilitator,
- the observer(s), and
- the recorder.

Each of these roles is described in the following sections.

Test Users

The test users (sometimes termed participants) are the people who sit at the computer and perform the defined tasks. Ideally, the test users should be the target end users, or people with similar backgrounds. If your pool of test users is large enough, you may wish to pair the test users to encourage speaking aloud during the tests. If you do choose to pair the test users, choose people with similar levels of experience to work together. When asking test users to participate in the test, emphasize that the purpose of the test is to test the *usability of the software*, not *their use* of the software.

Facilitator

The facilitator plays the role of the "host" of the usability test. Ideally, the facilitator should be a neutral party, with no vested interest in the results of the usability test. The facilitator welcomes everyone to the test, goes over the "rules" for the testing process, gives the tasks to the test users, and encourages the test users to "think aloud". The facilitator does not help or guide the test users. The exception is when the test users reach an impasse. In this case, once the associated usability problem has been observed, the facilitator can explain to the test users where the software let them down, how to complete the task, and then instruct the test users to move on to the next task.

Observers

Observation of real, representative end-users performing a task with the software gives a remarkably revealing picture of the software's usability. It is extremely beneficial to have the software developers and decision makers participate as observers. It is much more convincing and comprehensible to witness a usability flaw than to read about it in an evaluation report. A good mix of talent among the observers enriches the evaluation. For instance, an experienced HCI professional will provide judgments about what he/she has observed based on expert knowledge of design principles and guidelines.

The observer's role is to simply watch the test users as they perform the defined tasks and to keep notes on where the test users have difficulty using the system. **It is extremely important that the observers remain silent and expressionless throughout the test.** That means not talking, laughing, gesturing, or even rolling of eyes. Under no circumstances should any indication be given to the test user that his/her ability is being judged.

Recorder

The recorder observes the test user as the tasks are being performed and records the metrics. For example, if one of the metrics is time to perform, the recorder would have a timer and would time the test user as each task is performed and record those times. Like the observer, the recorder should remain silent and expressionless throughout the test. Appendix A has a sample form for recording this information.

3.2 Test Session Procedures

3.2.1 Testing Facility

The usability tests do not need to be run in a formal usability lab. The tests may be run in any area which is large enough to comfortably hold all of the test personnel without the test users being distracted by the observers. The area should also be free of other distractions (e.g., telephone rings, doors opening/closing). The test user(s) sits at the computer, with the facilitator and the recorder next to him or her. The observers sit behind the test user, where they can observe but be unobtrusive. One or more cameras may be used to record the actions of the test user(s) as well. The cameras can be simple video cameras (with audio) on tripods set to record the computer screen, the keyboard, and/or the test user. See Figure 1 for a sample room layout during a usability test.

3.2.2 Test Sessions

The length of the test sessions is determined based on the system being tested. For a typical system, the test sessions should be limited to one hour. It is easier to recruit test users when the testing session is short. A longer testing session would be useful if you wish to observe the effects of user fatigue on the usability of the system.

Prior to the test users arriving for the usability testing session, the facilitator, recorder, and observers take their seats. The facilitator reviews the rules for the observers and distributes note sheets for the observers and the recorder. (See appendix A for sample recording forms.)

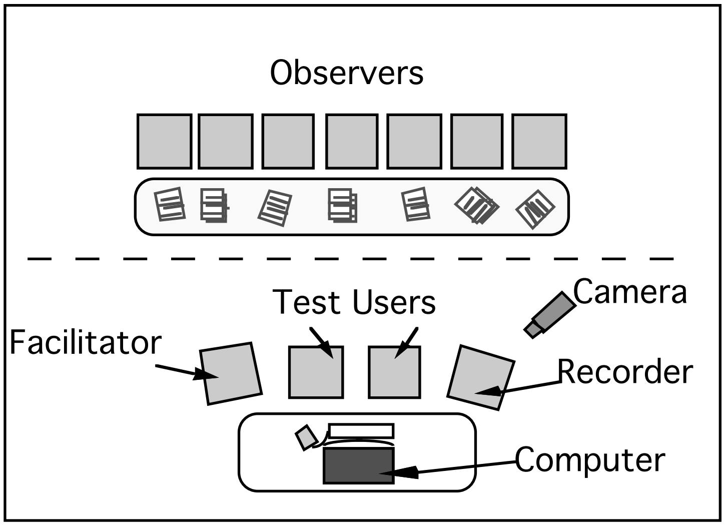


Figure 1. Sample Room Layout for Usability Test

When the test users arrive, the facilitator welcomes them and goes over the testing procedures. The test users are reminded that they are not being tested, the system is. If they have not done so earlier, the test users should sign release forms (see Appendix A). The release form is merely a formality which explains that the test users will be asked to perform some tasks with the software and to share their opinions, surprises and/or frustrations. The release form informs the test users that all of the collected data is confidential and that they have the right to stop participating in the test at any time. If the test is expected to be longer than an hour, they are told that they may take a break at any time.

The facilitator gives the tasks to the test user one at a time so that the test user does not know how many tasks there are and therefore does not feel that he/she isn't performing the tasks fast enough. The test user reads the task, asks any questions necessary to clarify the directions, and indicates to the facilitator when he/she is ready to begin the task. At this point the recorder should start timing the test user. The facilitator encourages the test user to talk aloud with stream of consciousness while proceeding through the tasks so that the observers can gain a better understanding of his/her thought process. After each task is completed, the test user is asked to give it a satisfaction rating indicating how easy or difficult the task was to perform.

When appropriate, the facilitator uses directed-dialog inquiries to keep the test users "thinking out loud" (e.g., "Is this what you were expecting? What are you looking for?" "What do you think the application is telling you?")

The observers make notes for each task performed. Observers are looking for the following problems:

- Where did the test user get misled?
- Where did the test user need to know something that he couldn’t possibly have known?
- Where did the test user succeed?
- Did the test user notice when he made a mistake? Could he recover? Could the mistake have been prevented?

The sessions can be video-taped, which helps during the analysis phase when trying to recall a particular problem from the usability test. One of the problems, though, with video recording is the volume of information collected and trying to locate the appropriate segment of tape that is desired during analysis. There are tools to help with this process, but we have not yet utilized any of them.

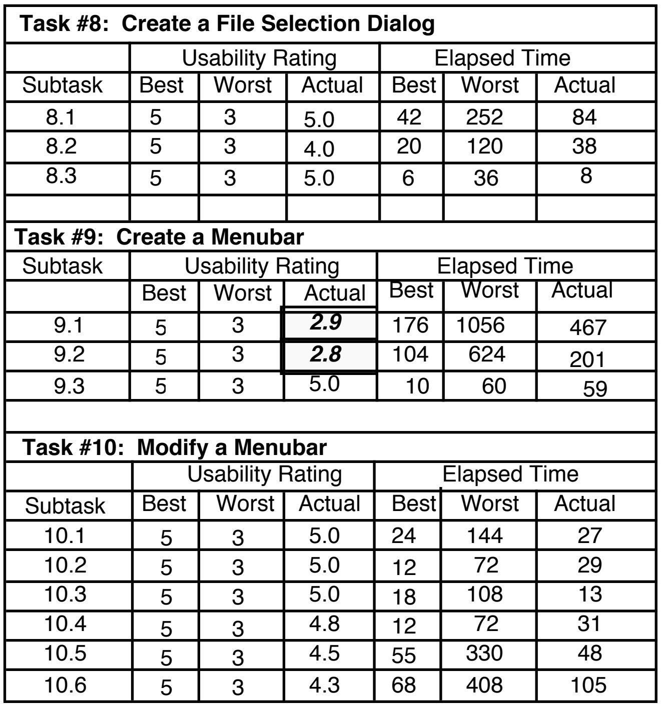
At the end of the test session, the test users are invited to give general comments about the system. The observers are then invited to ask any questions of the test users to clarify what they observed. The test users are then thanked and may be given a token of appreciation for taking the time to participate in the test.

4 What Do You Do With The Results From The Usability Testing?

Now that you have performed the usability testing and have collected all of this data, what do you do with it?

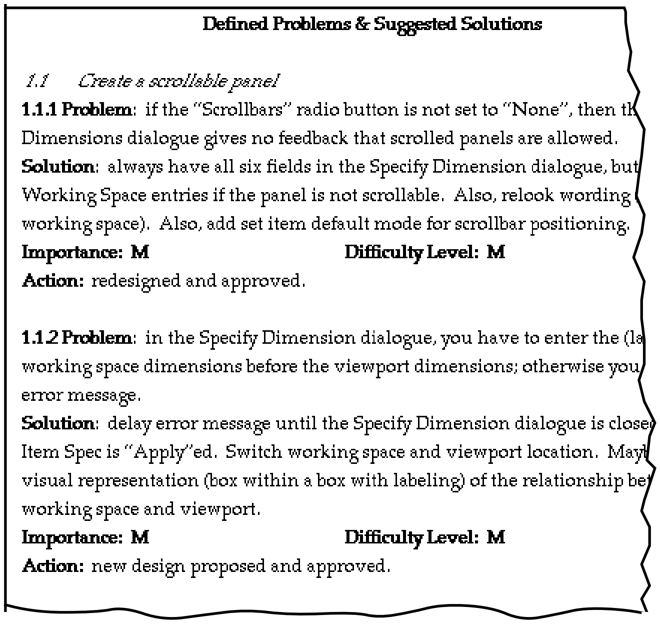
4.1 Test Metrics Compilation

The data collected from the usability tests must be compiled into a usable form. For each task, a mean score is determined from the scores collected from each test user and recorded in a Usability Test Metrics Table (Figure 2). If one test user's scores vary greatly from the others, a reason should be determined for this variation. If warranted, widely varying scores may be excluded from the mean calculation but be sure this fact is noted. (A "good" reason may be that there were unusual distractions during the test, for example.) A list of observed problems should be compiled from the observers' notes. Figure 3 shows a sample list of compiled problems.



Task #8: Create a File Selection Dialog						
	Usability Rating			Elapsed Time		
Subtask	Best	Worst	Actual	Best	Worst	Actual
8.1	5	3	5.0	42	252	84
8.2	5	3	4.0	20	120	38
8.3	5	3	5.0	6	36	8
Task #9: Create a Menubar						
Subtask	Usability Rating			Elapsed Time		
	Best	Worst	Actual	Best	Worst	Actual
9.1	5	3	***2.9***	176	1056	467
9.2	5	3	***2.8***	104	624	201
9.3	5	3	5.0	10	60	59
Task #10: Modify a Menubar						
	Usability Rating			Elapsed Time		
Subtask	Best	Worst	Actual	Best	Worst	Actual
10.1	5	3	5.0	24	144	27
10.2	5	3	5.0	12	72	29
10.3	5	3	5.0	18	108	13
10.4	5	3	4.8	12	72	31
10.5	5	3	4.5	55	330	48
10.6	5	3	4.3	68	408	105

Figure 2. Sample Usability Test Metrics Table



Defined Problems & Suggested Solutions

1.1 Create a scrollable panel

1.1.1 Problem: if the "Scrollbars" radio button is not set to "None", then th
Dimensions dialogue gives no feedback that scrolled panels are allowed.
Solution: always have all six fields in the Specify Dimension dialogue, but
Working Space entries if the panel is not scrollable. Also, relook wording
working space). Also, add set item default mode for scrollbar positioning.
Importance: M **Difficulty Level: M**
Action: redesigned and approved.

1.1.2 Problem: in the Specify Dimension dialogue, you have to enter the (la
working space dimensions before the viewport dimensions; otherwise you
error message.
Solution: delay error message until the Specify Dimension dialogue is close
Item Spec is "Apply"ed. Switch working space and viewport location. May
visual representation (box within a box with labeling) of the relationship be
working space and viewport.
Importance: M **Difficulty Level: M**
Action: new design proposed and approved.

Figure 3. Sample Evaluation Report

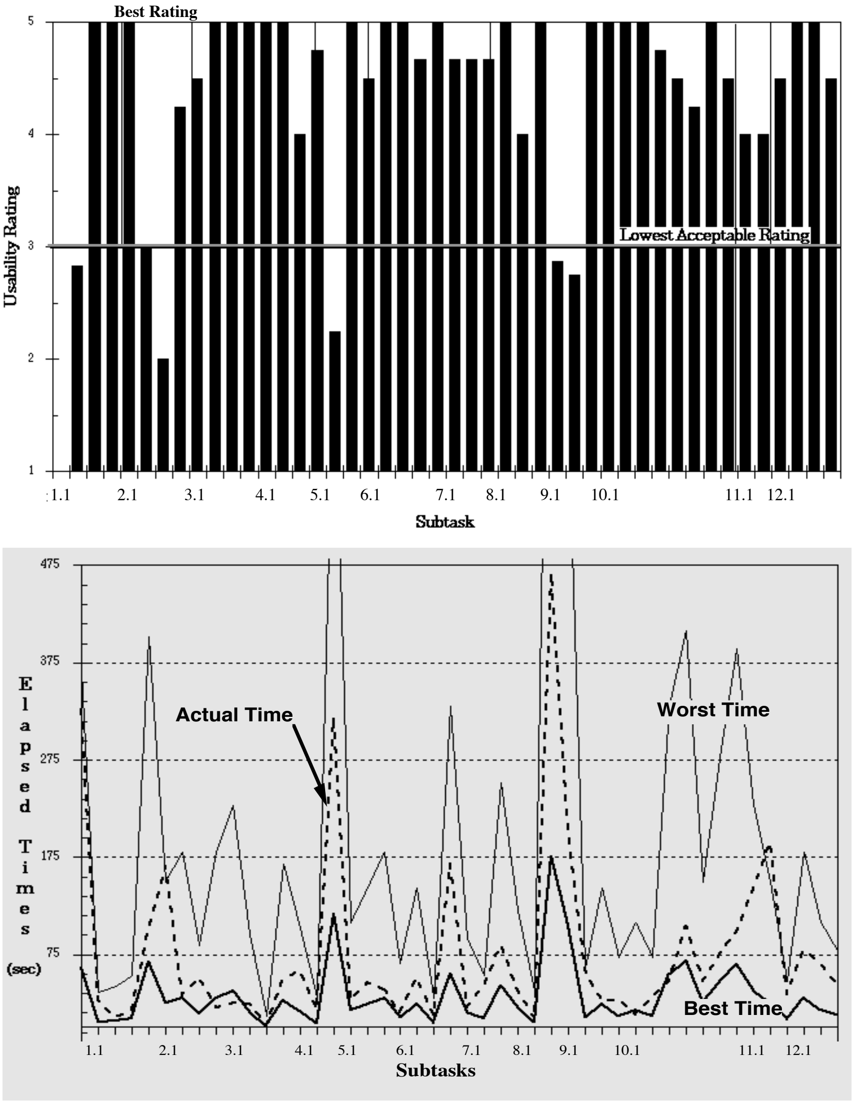
4.2 Data Analysis

Once the data is compiled, it is compared with the worst acceptable levels that were determined prior to the testing. Any task which has a time or usability rating (or other metric that was measured) outside of the "worst acceptable" range is flagged as having a serious usability problem. Plotting the data gives a quick way to visually identify the tasks with an unacceptable usability metric (Figures 4a & 4b).

Next, a meeting with the user interface designers and developers is held to go over the test results and review the observer's notes. Possible solutions to all problems found are proposed, whether the problem resulted in a "worst level" score or not. A list of proposed modifications is generated as a result of this meeting.

Next, each problem/modification is assigned an importance factor and an impact factor. The importance factor indicates how serious the problem is, with ratings as follows:

0 = low importance,
1 = medium importance, and
2 = high importance.



Figures 4a & 4b. Plotted User Satisfaction and Time to Perform Metrics

Any problem that resulted in a score below the "worst acceptable level" is automatically assigned a high importance factor. Each task modification is also assigned an impact factor which reflects how difficult the modification is to implement, as follows:

0 = easy to change,
1 = medium difficulty to change, and
2 = hard to change.

These impact factors should be associated with some development estimates. For example, easy-to-change means modification is expected to take less than one day, medium means it will take one to three days to change, and hard means it will take more than three days to change.

Once the importance and impact factors are assigned, a priority, or weighted cost, for fixes is determined as

priority = difficulty factor - importance factor + 2.

This results in priority values of 0 through 4 (see Figure 5).

		Importance Factor		
		low = 0	*med = 1*	*high = 2*
Difficulty Factor	*easy = 0*	2	1	0
	med = 1	3	2	1
	hard = 2	4	3	2

Priority of changes = difficulty factor - importance factor + 2

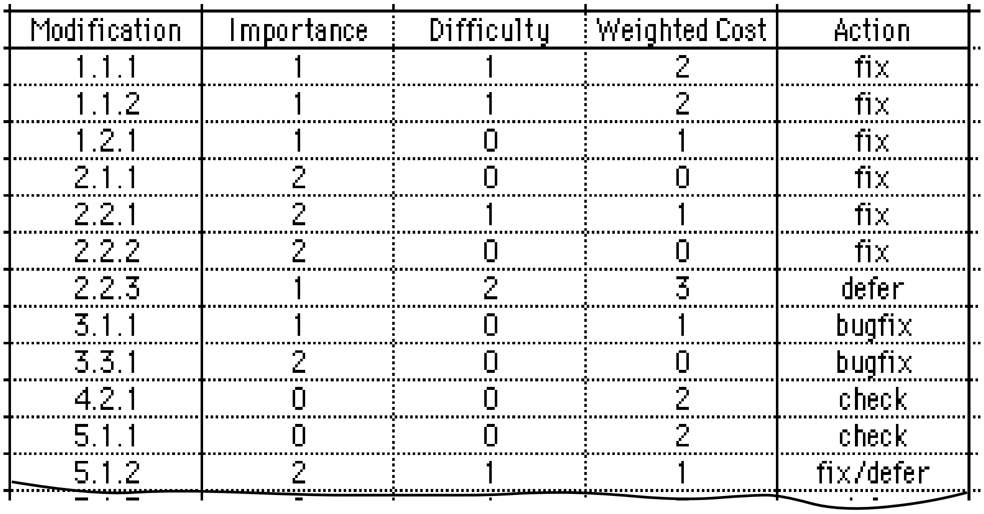
Figure 5. Priority Matrix

Next determine for each task whether the modification will be implemented and produce a summary report of the proposed modifications (Figure 6). All modifications with a priority rating of 0 should automatically be implemented, since these are high importance and easy to change. It is desirable to implement all the modifications with a priority rating of 0 to 2, to ensure that all of the problems deemed critical are fixed.

4.3 Follow-up Testing

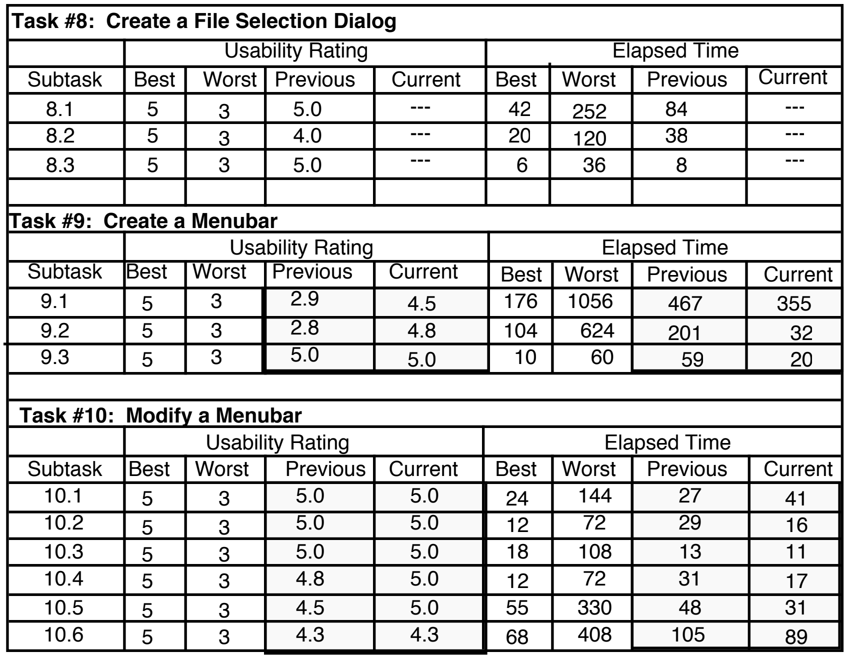
Once the modifications have been made to the system, it is good to do some follow-up usability testing to determine whether the changes fixed the problem, and to verify that they did not introduce new usability problems. It may not always be practical to re-test all of the tasks which resulted in modifications; in this case you should retest the tasks that revealed the most serious problems and those that resulted in the most dramatic changes.

If possible, the tasks should be retested with the test users that participated in the original usability tests, as well as new test users. The repeat test users can provide feedback on the changes made. They also provide a source of comparison, e.g., if the scores for the repeat test users are worse than the original test, then the problems were not fixed. The new test users, who should have similar experience to the original test users, provide another view of the usability of the improved interface.



Modification	Importance	Difficulty	Weighted Cost	Action
1.1.1	1	1	2	fix
1.1.2	1	1	2	fix
1.2.1	1	0	1	fix
2.1.1	2	0	0	fix
2.2.1	2	1	1	fix
2.2.2	2	0	0	fix
2.2.3	1	2	3	defer
3.1.1	1	0	1	bugfix
3.3.1	2	0	0	bugfix
4.2.1	0	0	2	check
5.1.1	0	0	2	check
5.1.2	2	1	1	fix/defer

Figure 6. Proposed Modification Summary Report



Task #8: Create a File Selection Dialog								
	Usability Rating				Elapsed Time			
Subtask	Best	Worst	Previous	Current	Best	Worst	Previous	Current
8.1	5	3	5.0	---	42	252	84	---
8.2	5	3	4.0	---	20	120	38	---
8.3	5	3	5.0	---	6	36	8	---
Task #9: Create a Menubar								
	Usability Rating				Elapsed Time			
Subtask	Best	Worst	Previous	Current	Best	Worst	Previous	Current
9.1	5	3	2.9	4.5	176	1056	467	355
9.2	5	3	2.8	4.8	104	624	201	32
9.3	5	3	5.0	5.0	10	60	59	20
Task #10: Modify a Menubar								
	Usability Rating				Elapsed Time			
Subtask	Best	Worst	Previous	Current	Best	Worst	Previous	Current
10.1	5	3	5.0	5.0	24	144	27	41
10.2	5	3	5.0	5.0	12	72	29	16
10.3	5	3	5.0	5.0	18	108	13	11
10.4	5	3	4.8	5.0	12	72	31	17
10.5	5	3	4.5	5.0	55	330	48	31
10.6	5	3	4.3	4.3	68	408	105	89

Figure 7. Sample Follow-up Usability Test Metrics Table

5 Summary

This document has presented the usability testing method which has been used in Code 522. Overall, this test procedure has worked smoothly despite the test users, facilitator, and observers all sharing the same room. Although some observers initially were unsure whether they could remain totally silent during the testing, they have successfully adhered to the non-distraction rule. Consequently, the test users have quickly forgotten about them. Observers and test users have given the testing process high marks.

Not yet included in this process is the tracking of the costs associated with preparing and conducting usability tests and calculating the cost benefits of usability testing. To calculate the costs of usability testing, we need to collect data on time spent during various phases of the usability testing process including times for: preparations (defining tasks, review/developer testing), the testing sessions, compilation of the results, the analysis of the results, the redesign and review of the changes, the implementation and testing of the changes and the retesting and re-analysis process.

To compute the financial benefits of usability testing, one is required to determine the value of the benefits from which the cost of the testing can be subtracted. It is tough to compute some of these benefits. Potential benefits associated with usability testing include increased productivity, increased sales or revenue, decreased training costs, decreased service costs, decreased user errors, decreased maintenance costs and increased user satisfaction. Unfortunately, it is much easier to measure and compute the cost of usability testing than to compute the cost benefits associated with the testing. This helps to explain why so many managers are inclined to view usability engineering activities as “all cost, no [obvious] benefit.” They need cost benefit figures to justify the required investment in the usability testing.

Once we have defined the methods for collecting the cost and calculating the benefits associated with performing usability tests, we will update this handbook to incorporate this data collection into our usability testing procedures.

6 Suggested Readings

The following is a listing of sources of further information on usability which are available in Code 588.

Usability Professionals' Association

The Usability Professionals Association publishes a newsletter called Common Ground. The newsletter contains articles related to usability and lists upcoming conferences and recent articles printed in other journals. For more information visit their web site at

http://www.upassoc.org
or contact the UPA office at
414 Plaza Drive, Suite 209
Westmont, IL 60559
Phone: 630-655-1647
Fax: 630-655-0391

Adler, Paul S. and Terry A. Winograd. Usability: Turning Technologies into Tools. Oxford University, 1992.

Hix, Deborah and H. Rex Hartson. Developing User Interfaces: Ensuring Usability Through Product and Process. John Wiley & Sons, Inc., 1993.

Nielsen, Jakob. Usability Engineering. Academic Press, Inc. 1993.

This book discusses the usability engineering lifecycle including usability testing. This book also contains an extensive bibliography.

Norman, Donald A. The Design of Everyday Things. Doubleday. 1990.

This book was previously published with the title The Psychology of Everyday Things. It discusses general usability issues of everyday objects such as doors and phones. Useful and fun guide to general usability issues.

Macleod, Miles. An Introduction to Usability Evaluation. National Physical Laboratory; Teddington, Middlesex, UK. 1992

References

The following documents are cited in the text.

1. Boehm, B., "Industrial Software Metrics Top Ten List", IEEE Software, September 1987

2. Fairley, M. E. "Design and code inspections to reduce errors in program developments", IBM System Journal, 3, 1976

3. Hakiel, Simon, “CICS/ESA Usability: A Measure of Success”, People and Computers VII, HCI’92 Conference Proceedings, British Computer Society Conference Series 5, September 1992

4. Human Factors Group, "The Human Factors Group at Compaq Computer Corporation", CHI'92 Conference Proceedings, May, 1992

5. Jeffries, R. et al., "User Interface Evaluation in the Real World: A Comparison of Four Techniques", CHI'91 Conference Proceedings, April, 1991

6. Macleod, Miles, "An Introduction to Usability Evaluation", NPL Report DITC 102/92, National Physical Laboratory, Teddington, Middlesex, U.K.

Bibliography

The following documents are not specifically referenced in the text but may provide additional information on document formatting and production.

Hartson, H. Rex, "Developing Usable Interfaces for Real Interactive Systems", ACM Professional Development Seminar for Washington DC. Chapter of the ACM, March, 1992

Mrazek, D. & Rafeld, M., "Integrating Human Factors on a Large Scale: Product Usability Champions", CHI'92 Conference Proceedings, May, 1992

Spool, Jared, "Product Usability: Survival Techniques", Tutorial given at CHI'92 Conference, May, 1992

Szczur, Martha, "NASA's TAE Plus: a GUI Development Tool and Application Environment", The X Resource Journal, Issue 2, Spring 1992

Szczur, Martha, "Usability Testing on a Budget: A NASA Usability Test Case Study", Behaviour & Information Technology, 1994, Vol. 13, No. 1

Appendix A Sample Forms

This appendix contains sample forms that are used during the usability testing process. Electronic versions of these forms are available on the Code 520 Macintosh server.

A-1 Sample Recorder Form

Task 1: Create a New Evaluation		Usability Rating			Elapsed Time		
Subtask	Subtask name	Best	Worst	Actual	Best	Worst	Actual
1.1	Open a new evaluation						
1.2	Classify eval as a minor weakness						
1.3	Indicate weakness is correctable						
1.4	Enter a question for the offeror						
1.5	Enter a document reference						
1.6	Assign assessment criterion						
1.7	Enter descriptive title						
1.8	Enter details text						
1.9	Enter impact if weakness not corr						
1.10	Save the evaluation						
1.11	Locate new evaluation on browser						

Task 2: Report on an Evaluation		Usability Rating			Elapsed Time		
Subtask	Subtask name	Best	Worst	Actual	Best	Worst	Actual
2.1	Locate an evaluation in the brows						
2.2	Generate detailed report of eval...						
2.3	print the report						

Task 3: Edit an Evaluation		Usability Rating			Elapsed Time		
Subtask	Subtask name	Best	Worst	Actual	Best	Worst	Actual
3.1	Open an evaluation						
3.2	minimize the AES application						
3.3	open AES library document						
3.4	copy section of text						
3.5	switch to the AES application						
3.6	append copied text to "Details" fie						
3.7	save changes to the evaluation						
3.8	close the evaluation form window						
3.9	submit eval to committee chair						

Task 4: Move and Archive an Evaluation		Usability Rating			Elapsed Time		
Subtask	Subtask name	Best	Worst	Actual	Best	Worst	Actual
4.1	Filter the evaluation tree						
4.2	Locate evaluation						
4.3	Move the evaluation to another are						
4.4	Archive selected evaluation						

A-2 Sample Release Form

Statement of Informed Consent

You are asked to participate in an study of how SEB members use the AES. The purpose of the study is to give us information we need to make design decisions for the user interface.

In the study we will ask you to perform some tasks using the AES. The tasks have been designed to use a range of features of the AES. We are not evaluating you, rather we are studying how easy the AES is to use. All information that you give us and all data that we collect concerning your task performance will be held in strict confidence. We will use the information for statistical and summary purposes only, and will make certain that your name is not associated with your records.

The facilitator will assist you and answer any questions you have. You are completely free to stop participating in the software evaluation at any time.

If you are willing to participate, please sign the following statement:

> "I have read the above description of the software evaluation procedure and of my rights as a participant and I have agreed to take part in the evaluation on the ease of use of the AES.
>
> Signed:
>
> Date:

A-3 Facilitator Notes

- Remind Observers

  -- read guidelines, make comments on forms

  -- no talking, snorting, laughing allowed -- silent observer

  -- testers are not being tested, software is being tested

- Remind Participants

  -- think out loud (talk to each other)

  -- stop at any time

  -- no set number of tasks to be performed -- just keep going until time limit

  -- after each task, give a user satisfaction rating between 1-> 5
  (1= difficult --> 5 = easy)

  -- if tester has identified a problem that prevents them from completing task in reasonable time, tester or facilitator can decide to move on to next task; at this time, facilitator can help participants finish task to get a sense of completion

  -- okay to ask questions of facilitator

  -- indicate verbally when ready to start a task (after reading instructions)

  -- don’t try to rush... you are not being tested ... the software is being tested

- Remember, as facilitator, to use directed-dialog inquiries --

  -- Is this what you expected?
  -- What are you looking for?
  -- What do you think it is telling you?
  -- Does documentation help?

A-4 Usability Test Checklist

- Tester Consent Forms
- Tester's General Information Questionnaire
- Facilitator Notes
- Tester Tasks (one task per page)
- Observer Briefing Notes
- Observer Notes Worksheet
- Timer/Metric keeper Worksheet
- Stopwatch
- Video camera (if archiving test)
- "Test in Progress" signs
- Evaluation Form on Usability Process (if appropriate)
- Reward/Thank you treat for testers